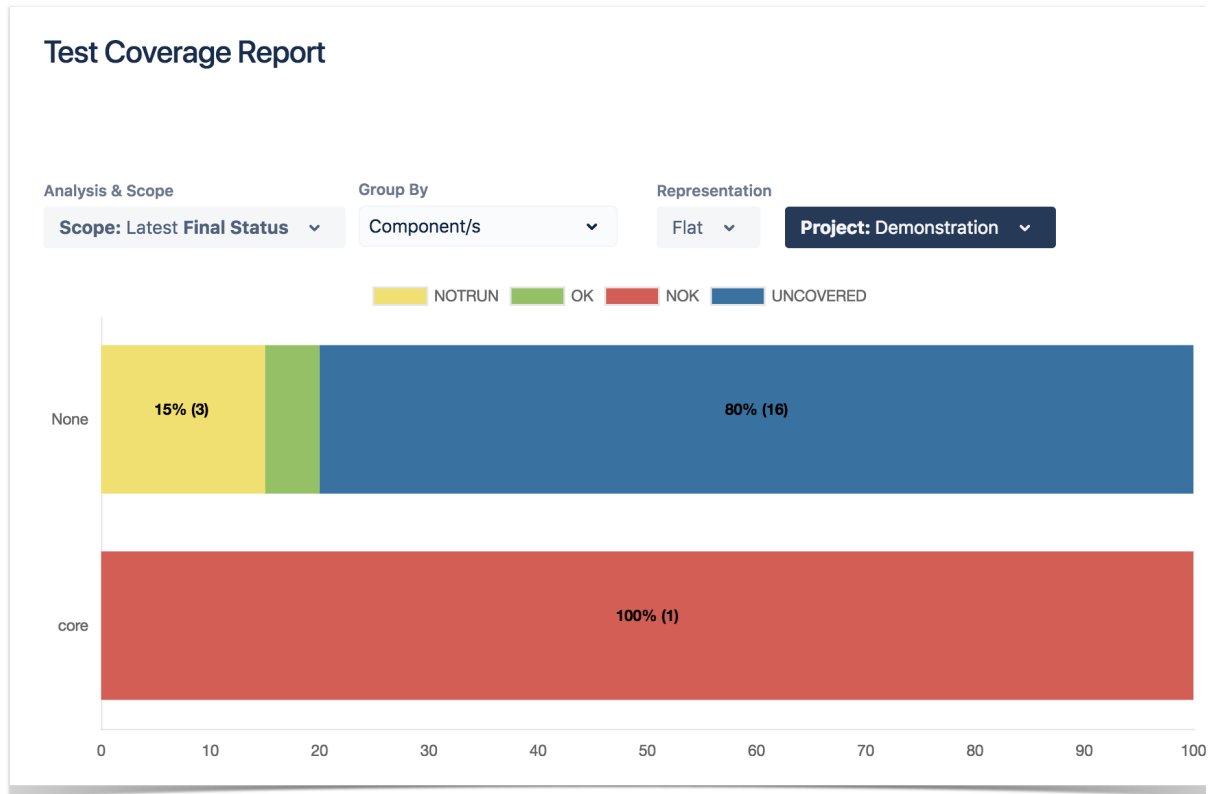


Cloud Coverage Report

1

Coverage Report

Calculates the test coverage status for a list of coverable issues for the selected version or Test Plan, and/or Test Environment. The statuses are presented in a normalized horizontal stacked bar chart and evaluates the completeness of each issue.



2

Purpose

In order to analyze the current status of your release, or even past releases, you need to evaluate the status of the requirements and other covered issues, taking into account the respective results.

You might want to analyze just the requirements implemented in a given version or you may want to analyze requirements from earlier versions based on your regression testing.

The Test Coverage report provides the means to analyze your requirements statuses in real-time.

3

How to use

This report is accessible either from the **Xray** icon on the left sidebar of the project or from the standard **Reports** icon, which includes other kind of reports besides Xray.

At the top of the report, you have some options that need to be configured.

- **Analysis & Scope:** how to analyze the issues
- **Group By:** how to visually group the coverable issues (e.g. by priority, by component)

- **View:** flat or hierarchical
- **Filters:** the source data for the report (e.g. the requirements or other covered issues) and how to evaluate the status for each one of the item being shown.



4

Source data / Filters

The source data for the report (e.g., the requirements or other covered issues) can be defined under **Filters**, which is available at the top left-hand corner.

Filters

Project: Bookstore

Basic | **Advanced**

Project: Bookstore	Assignee: Select...
Component: Select...	Status: Select...
Fix versions: Select...	Resolution: Select...
Saved Filter: Select...	Contains: <input type="text"/>

More

Clear | Reset to default | **Apply** | Cancel

Source "requirement"/coverable issues can be directly provided using a Basic or Advanced filter:

- **Advanced:** a JQL query to filter the coverable issues
- **Basic:**
 - **Project:** project
 - **Assignee:** the issue's assignee
 - **Component:** the component assigned to the issue
 - **Fix Versions:** versions assigned to the issue
 - **Resolution:** the workflow resolution
 - **Status:** the workflow status
 - **Contains:** the full issue key or part of the summary

Analysis & Scope

5

On the left side, you can define the analysis strategy, i.e., the way you want to analyze the selected/filtered requirements. You can choose either to analyze by **Version** or **Test Plan**, and then complement it with a **Test Environment**.

If you choose analysis by **Version**, then the latest results from Test Executions for the specified version are taken into account. If you choose **Test Plan**, only Test Executions (and related results) associated with the given Test Plan are considered during the calculation of the requirement status.

If you specify the Test Environment, then only the executions that ran in the specified environment are considered.

Analysis & Scope

Group By

Scope: Latest Final Status

Component/s

Calculate the Test Coverage for the following scopes.

Latest Version Test Plan

Test Environment

All Environments

☒ Final statuses have precedence over non-final.

Clear Apply Cancel

6

Group By

You can visually group the requirements by Priority, Component or other fields, so you can analyze requirements from different perspectives.



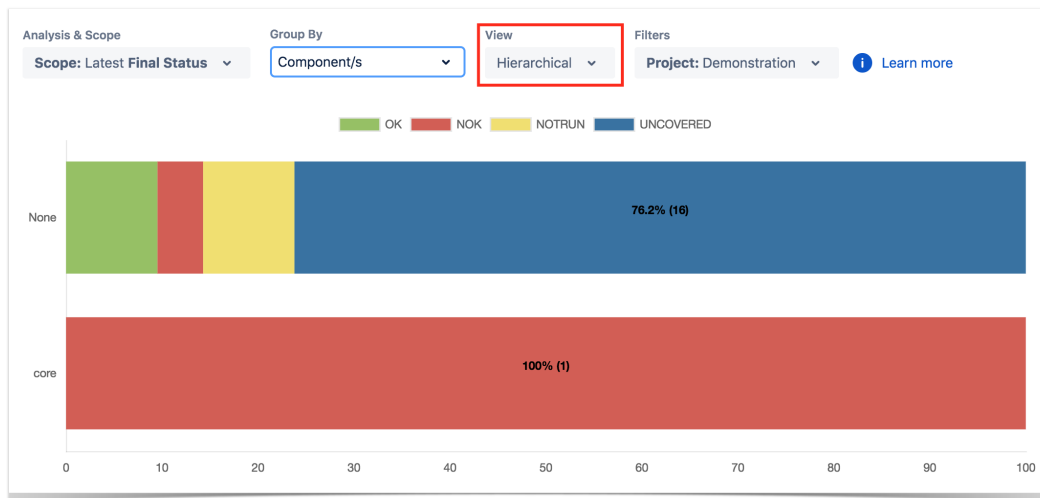
7

View

You can see a flattened list of coverable issues or just the top-level ones (e.g., the top-level parent requirements).

For example, if you're using Epics and Stories and you choose the "Flat" view, then only the Epics will appear in the report, independent of whether the source data includes the sub-issues or not. If you choose the "Hierarchical" view, then only the Epics would be shown.

- **Hierarchical** - only the parent coverable issues will be presented in the chart. Child issues can still be visualized in the overall details table.
- **Flat** - the parent/child relationship between coverable issues is not considered. All parent and child issues will be considered for the chart.



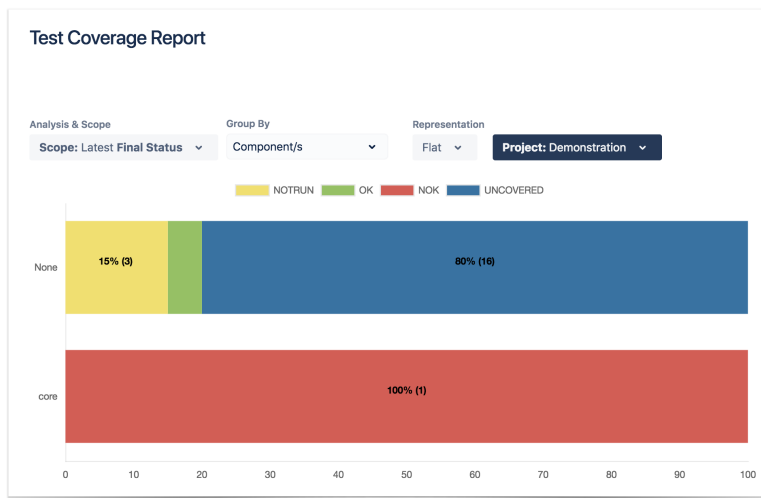
8

Example

Each bar on the chart represents a group of Requirements with a different *value* provided by the grouping field. Stacked within each bar are the Requirements grouped by coverage status: **OK**, **NOK**, **NOT RUN**, **UNKNOWN** and **UNCOVERED**.

Because this is a normalized chart, the bars are always 100% wide.

The Overall Coverage requirements chart is interactive. Hovering over each bar shows a small popup with the information on the specific series (Status, Grouping Field and Percentage). You can deactivate a particular series by clicking it in the series legend



The chart also supports drill-down. When a section is selected, a table appears below the chart with the Requirement issues.

For each Requirement issue, the table shows the following indicators:

- **Total Tests:** total number of linked tests
- **Passed:** total number of tests passing (i.e., the ones whose status is mapped to the Test Coverage status "OK")
- **Failed:** total number of tests failing (i.e., the ones whose status is mapped to the Test Coverage status "NOK")
- **Other:** total number of tests whose current status is mapped to the Test Coverage statuses "Not Run" or "Unknown"

It also contains a column with a progress bar that shows the completeness of each issue.

Some values may provide a link, so you can be redirected to the respective entities corresponding to the number shown.

Requirements with Component/s "core" and Status "NOK"

[View Issues](#)

Search

Key	Summary	Total Tests	Passed	Failed	Other	Completeness
DEMO-87	As a user, I can calculate the su...	4	2	1	1	<div><div>50%</div></div>



To better understand how coverage works, we recommend you to check the ["Understanding Coverage"](#) module.